



Optical Network as a Service for Service Function Chaining across Datacenters

Mehmeri, Victor; Wang, Xi; Zhang, Qiong; Palacharla, Paparao; Ikeuchi, Tadashi; Tafur Monroy, Idelfonso

Published in:
Optical Fiber Communication Conference 2017

Link to article, DOI:
[10.1364/OFC.2017.W4J.4](https://doi.org/10.1364/OFC.2017.W4J.4)

Publication date:
2017

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Mehmeri, V., Wang, X., Zhang, Q., Palacharla, P., Ikeuchi, T., & Tafur Monroy, I. (2017). Optical Network as a Service for Service Function Chaining across Datacenters. In *Optical Fiber Communication Conference 2017* Optical Society of America. <https://doi.org/10.1364/OFC.2017.W4J.4>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Optical Network as a Service for Service Function Chaining across Datacenters

Victor Mehmeri^{1,2}, Xi Wang¹, Qiong Zhang¹, Paparao Palacharla¹, Tadashi Ikeuchi¹, Idelfonso Tafur Monroy²

¹Fujitsu Laboratories of America, Inc. 2801 Telecom Pkwy, Richardson, TX, USA, Email: vime@fotonik.dtu.dk

²Technical University of Denmark, Department of Photonics Engineering, Ørsted Plads, Building 343, Kgs. Lyngby 2800, Denmark

Abstract: We present the SPN OS, a Network-as-a-Service orchestration platform for NFV/SDN integrated service provisioning across multiple datacenters over packet/optical networks. Our prototype showcases template-driven service function chaining and high-level network programming-based optical networking.

OCIS codes: (060.4253) Networks; (060.4510) Optical Communications

1. Introduction

Network virtualization has enabled the deployment of individual tenant-managed network services over a shared physical infrastructure. This growing trend facilitates the emergence of new business models such as Network-as-a-Service (NaaS), wherein infrastructure management is decoupled from service management to achieve maximized flexibility and improved resource utilization efficiency. On the other hand, network virtualization raises new issues, such as challenges in efficient provisioning of end-to-end services, scalable management and control of virtual and physical resources, independent programmability of virtual networks, and reliable traffic isolation in multi-tenant deployments.

In the meantime, there is growing need for comprehensive virtual network solutions that provide built-in Network Functions Virtualization (NFV) and Service Function Chaining (SFC) capabilities. In addition, the network functions contained in these services may span across multiple datacenters (e.g., NFV infrastructure such as central offices with servers/storage, customer/enterprise sites with single/few servers) interconnected by packet and/or optical networks for cost/performance optimization, and therefore require end-to-end networking as well as service function chaining. Currently such network service deployments require multiple controllers to manage each segment of the NFV/SDN service, thereby making it difficult to provide end-to-end per-tenant service orchestration without an integrated platform.

In this paper, we present a NaaS orchestration platform called the Software-Programmed Networking Operating System (SPN OS) for NFV/SDN integrated service provisioning over packet/optical networks. We also report a running prototype of the SPN OS demonstrating key features including multi-datacenter dynamic NFV/SDN service deployment and high-level network programming-driven packet and optical network control/management.

2. Overview of the SPN OS

The SPN OS implements Object-Oriented Network Virtualization (OONV) [1], which decouples distributed service control from centralized resource management, to offer individual tenants the flexibility of managing their end-to-end services over shared packet/optical network infrastructure. As shown in Fig. 1, a service is operated as a dedicated virtual network object (VNO) in the SPN OS, which provides a complete representation of the service throughout its lifecycle with built-in methods and operations. The network-wide resources are managed by a centralized Arbiter, which allocates the required resources upon VNO instantiation, and then delegates the actual control of the allocated resources to the instantiated VNOs. The VNO service representation includes both the network plane and the service plane. Programming of the VNO network plane is achieved with the high-level

network programming language NetKAT [2], which compiles high level network policies into packet/optical device-level SDN commands (e.g., OpenFlow entries), effectively enabling a “Program & Compile” model of networking [3] as opposed to the more traditional “Define & Configure” model. As for the VNO service plane, we employ ETSI NFV-MANO TOSCA [4] template and Network Service Headers (NSH) [5] for virtual network function (VNF) definition and SFC configuration.

A tenant interacts with its VNO(s) for various service/network operations by calling the VNO’s built-in functions through its northbound interface. Upon VNO activation, the VNO compiles its network programs to configure the packet/optical network for establishing its virtual network, and

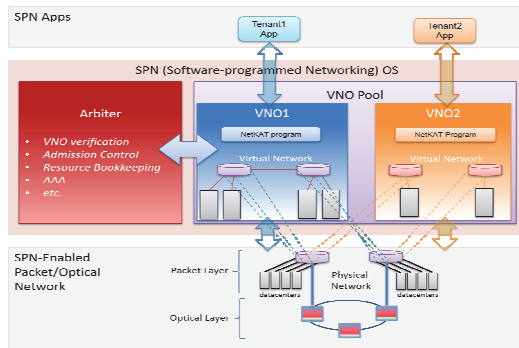


Fig. 1. SPN OS architecture overview.

then instantiates the required VNFs in the datacenter(s) and also configures the SFC for establishing the end-to-end NFV service. Since each VNO is running as an independent process, the outage of the Arbiter does not impact the service operation of running VNOs, such as reconfiguration of SFCs, packet routes and packet-to-optical mappings, etc. (The outage of the Arbiter only affects the operations that involve global resource management and allocation, such as VNO creation, deletion and migration).

3. SPN OS Prototype Implementation for SFC over Packet/Optical Networks

We built a prototype of the SPN OS to verify the following capabilities: (1) Centralized management of network-wide resources; (2) Distributed control of network based on high-level program compilation as opposed to direct data-plane configuration; (3) Seamless virtualization of hybrid packet and optical networks for providing end-to-end Connectivity as a Service; (4) End-hosts management for creating a service plane overlay on which tenants can deploy VNFs and establish dynamic Service Function Chains.

We tested our prototype on an emulation testbed as illustrated in Fig. 2. We used Mininet [6] for network emulation, Open vSwitch as virtual packet switches and LINC-OE [7] to emulate optical nodes, i.e., Reconfigurable Optical Add-Drop Multiplexers (ROADMs). OpenFlow 1.3.4 with optical extensions was used for configuring the LINC-OE optical nodes. The SPN OS consisted of a Java based application for both the Arbiter and each Tenant/VNO application, as well as a NetKAT compiler with a northbound REST interface. For SFC we customized an open-source development of OpenDaylight controller supporting NSH-based SFC to make it interwork with our Mininet dataplane. We programmed a VNO to establish a connectivity between two remote datacenters (each datacenter represented by a virtual packet switch and a server cluster) using the optical path shown in blue in the lower part of Fig. 2 (a). Also programmed was an alternative optical path (shown in green) to be used upon any failure notification that affects the original path, thus programming optical layer resiliency into the VNO.

Fig. 2 (a) gives an overview of the VNO including its Service Plane (as realized by the use of NSH headers with VxLAN), its virtual network layer and the physical layer it is deployed on. Also shown are the information contained in the VNO, such as the VNO's program and virtual layer, as well as the virtual-to-physical mapping information embedded by the Arbiter after resource allocation (this mapping information may be hidden from the tenant depending on the VNO's trust/security setting). The desired network behavior was programmed with a few dozen lines of NetKAT code (a pseudo-code snippet of which is shown in Fig. 2 (a) under "VNO Policy") and verified with ping test (for network connectivity) and HTTP traffic (for SFC connectivity). For testing SFC, we look at the main fields of the NSH headers which are the Network Service Path (NSP), which identifies the chain, and the Network Service Index (NSI) which identifies the current node (hop) within the chain and gets decremented at every hop. Fig. 2 (b) shows NSH header information as dumped by the VNFs upon arrival of packets travelling through the chain, as well as OpenFlow flows related to NSH matches and the successful establishment of HTTP traffic, steered through the chain by an SFC classifier. Since the chain deployed was bidirectional, we see two NSP values, 1063 and 8389671, for the forward and reverse paths, respectively. Average VNO operation time over 5 experiment runs is shown in Fig. 3 (a), corresponding to:

